# MEDICAL DIAGNOSIS

**A MINI PROJECT REPORT**

**18CSC305J – ARTIFICIAL INTELLIGENCE**

***Submitted by***

**Sahil Satasiya (RA2011026010110)**

**Varun Khachane (RA2011026010072)**

**Shresth Gupta (RA2011026010091)**

Under the guidance of

**Dr. M.S. Abirami**

Associate Professor, Department of Computer Science and Engineering

***In partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**


S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2023**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **"MEDICAL DIAGNOSIS"** is the bona fide work of **SAHIL SATASIYA (RA2011026010110), VARUN KHACHANE (RA2011026010072), SHRESTH GUPTA (RA2011026010091)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| Dr. M.S. Abirami | Dr. R.Annie Uthra |
| **GUIDE** | **HEAD OF THE DEPARTMENT** |
| Associate Professor | Professor & Head |
| Department of Computing Technologies | Department of Computational Intelligence |

# TABLE OF CONTENT

# ABSTRACT

Identifying and categorising diseases or conditions based on patient symptoms, medical history, and diagnostic testing is a difficult procedure in medicine. For successful treatment and better patient outcomes, a prompt and accurate diagnosis is essential. The goal of this project is to create a medical diagnostics system that leverages artificial intelligence (AI) methods to improve diagnostic precision and effectiveness.

An extensive dataset of patient information, including symptoms, medical records, test results, and imaging data, is collected and analysed as the project's first step. This dataset serves as the basis for developing and testing AI diagnostic algorithms.

To discover patterns and connections between patient attributes and diagnoses, machine learning methods like as decision trees, support vector machines, or deep learning models are used to the dataset. To enhance their diagnostic capabilities and generalisation performance, these models go through training and optimisation techniques.

Healthcare practitioners can enter patient data into the built medical diagnostics system and receive diagnostic predictions or recommendations thanks to its user-friendly interfaces. In order to analyse the input data, compare it to recognised patterns, and produce accurate and trustworthy diagnoses, the system makes use of AI algorithms.

By contrasting the diagnostic accuracy, sensitivity, specificity, and other pertinent metrics of the medical diagnosis system with currently used diagnostic techniques or professional judgements, its performance is assessed. Clinical validation studies and user input are used to determine how well the system works in supporting medical practitioners in making diagnostic judgements.

This project's importance stems from its potential to improve medical diagnosis through the use of AI tools. By offering evidence-based advice, lowering diagnostic errors, and enabling earlier detection of diseases or ailments, the proposed system can help healthcare providers. The initiative makes a contribution to the burgeoning field of AI-assisted healthcare, where cutting-edge technologies are used to raise patient care and diagnostic precision.

In conclusion, this project aims to develop a medical diagnosis system using AI techniques to improve diagnostic accuracy and efficiency. By analyzing patient data, learning from patterns, and providing diagnostic predictions, the system offers valuable support to healthcare professionals in making informed diagnostic decisions. The integration of AI in medical diagnosis holds immense potential to advance healthcare practices and improve patient outcomes.

# CHAPTER 1

## INTRODUCTION

The identification and determination of diseases, ailments, or disorders in persons based on their symptoms, medical history, physical examination, and diagnostic testing is a fundamental and crucial element of healthcare. It is essential for directing sensible therapy choices and enhancing patient outcomes.

Healthcare experts, such as doctors, specialists, and diagnosticians, are needed to perform the process of diagnosing a medical condition. These professionals use their clinical competence, medical knowledge, and other resources to make an accurate diagnosis. Clinical reasoning, pattern identification, and cognitive abilities of the doctor have traditionally been significantly weighted in medical diagnosis.

Artificial intelligence (AI) and machine learning algorithms, on the other hand, have been progressively incorporated into medical diagnostics as a result of technological improvements and the emergence of computational methodologies. Through the analysis of enormous volumes of patient data, the discovery of intricate patterns, and the provision of evidence-based insights to aid healthcare professionals in their decision-making, these technologies have the potential to complement and enhance the diagnostic process.

The effective analysis of patient information, including symptoms, medical history, test results, and imaging data is made possible by the use of AI in medical diagnosis. Machine learning algorithms can gain knowledge from this data and create models that can identify relationships, categorise diseases, forecast results, and make suggestions for additional research or treatment alternatives.

Healthcare practitioners can benefit from AI-based medical diagnosis systems by getting a second opinion, cutting down on diagnostic mistakes, increasing diagnostic precision, and facilitating illness early detection. By taking into account a variety of patient data, including uncommon or subtle patterns that could be difficult for human diagnosticians to spot, they might offer insightful information.

Additionally, a medical diagnosis is not only applicable to certain patient circumstances. It also includes epidemiological investigations, population-level analyses, and the recognition of public health issues. AI algorithms can help in tracking the transmission of infectious diseases, identifying disease outbreaks, and developing preventive measures and interventions.

It is crucial to remember that AI-based medical diagnosis systems should always be used as a tool to support healthcare practitioners rather than taking the place of their knowledge. They should be developed and validated using solid clinical data. To ensure patient safety, data security, and responsible deployment, the integration of AI should be led by ethical concerns, privacy protections, and regulatory frameworks.

# CHAPTER 2

## LITERATURE SURVEY

1. "Deep learning in medical imaging: A comprehensive review" by Litjens et al. (2017): This review provides an overview of the applications of deep learning in medical imaging, including image classification, segmentation, and detection. It discusses the advantages and challenges of using deep learning algorithms for medical diagnosis and highlights the potential impact of these techniques in improving diagnostic accuracy.

2. "Machine learning for medical diagnosis: history, state of the art and perspective" by Raja et al. (2019): This paper presents a comprehensive survey of machine learning techniques applied to medical diagnosis. It covers various machine learning algorithms, including decision trees, support vector machines, and neural networks, and their applications in different medical domains. The paper also discusses the challenges and future directions in the field.

3. "Artificial intelligence in healthcare: Past, present and future" by Krittanawong et al. (2020): This review article provides an overview of the historical development, current state, and future prospects of artificial intelligence in healthcare. It discusses the applications of AI in medical diagnosis, clinical decision support, and patient management. The paper also addresses the ethical considerations, challenges, and potential impact of AI on healthcare systems.

4. "Medical diagnosis using machine learning: A review" by Gopalakrishnan et al. (2019): This review summarizes the application of machine learning techniques for medical diagnosis across various medical specialties. It discusses the use of different algorithms and their performance in terms of accuracy and efficiency. The paper also highlights the challenges, limitations, and future directions in the field.

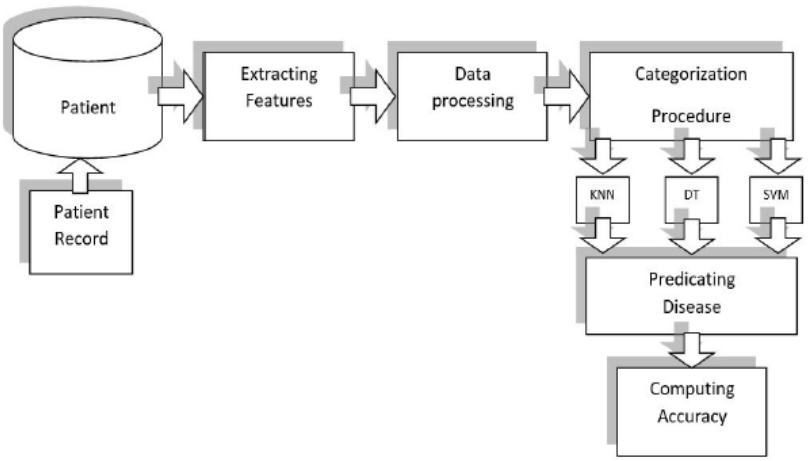
5. "Artificial intelligence in medicine: Current trends and future possibilities" by Char et al. (2017): This paper provides an overview of the current trends and future possibilities of artificial intelligence in medicine. It discusses the use of AI techniques in medical diagnosis, personalized medicine, and healthcare management. The paper also addresses the challenges and potential impact of AI on the healthcare industry.

6. "Machine learning and medical imaging" by Shen et al. (2017): This review focuses on the application of machine learning techniques in medical imaging for disease diagnosis and prognosis. It discusses various imaging modalities, such as magnetic resonance imaging (MRI), computed tomography (CT), and positron emission tomography (PET), and how machine learning algorithms can analyze and interpret these images to assist in medical diagnosis.

7. "Artificial intelligence in healthcare: A systematic review" by Rajkomar et al. (2019): This systematic review explores the applications of artificial intelligence in healthcare, including medical diagnosis, prediction, and decision-making. It provides insights into the current state of AI technologies, their limitations, and potential implications for clinical practice.

# CHAPTER 3

## SYSTEM ARCHITECTURE AND DESIGN




The system architecture and design for medical diagnosis involve the integration of various components, algorithms, and data sources to create an efficient and accurate diagnostic system.

Here is a general outline of the system architecture and design considerations for medical diagnosis:

1. **Data Collection and Integration:**

    - Identify and collect relevant patient data, including medical history, symptoms, laboratory results, and imaging data.
    - Integrate data from multiple sources, such as electronic health records (EHRs), medical devices, and external databases.

2. **Data Preprocessing and Feature Extraction:**

    - Clean and preprocess the collected data to ensure data quality and consistency.
    - Extract relevant features from the data using techniques such as signal processing, image analysis, or natural language processing (NLP).

3. **Data Storage and Management:**

    - Design a robust database system to store and manage the collected patient data.
    - Ensure data privacy, security, and compliance with healthcare regulations, such as HIPAA.

4. **Machine Learning and AI Algorithms:**

    - Select appropriate machine learning and AI algorithms based on the nature of the medical diagnosis problem.
    - Train the algorithms using labeled data to learn patterns and relationships between input features and diagnostic outcomes.
    - Consider various algorithms, such as decision trees, support vector machines, neural networks, or deep learning models, based on the complexity and requirements of the medical diagnosis task.

5. **Validation and Evaluation:**

    - Validate the performance and accuracy of the diagnostic system using independent datasets or expert opinions.
    - Evaluate the system's sensitivity, specificity, positive predictive value, and other relevant metrics.
    - Conduct clinical validation studies to assess the system's impact on healthcare outcomes and patient care.

6. **Iterative Improvement:**

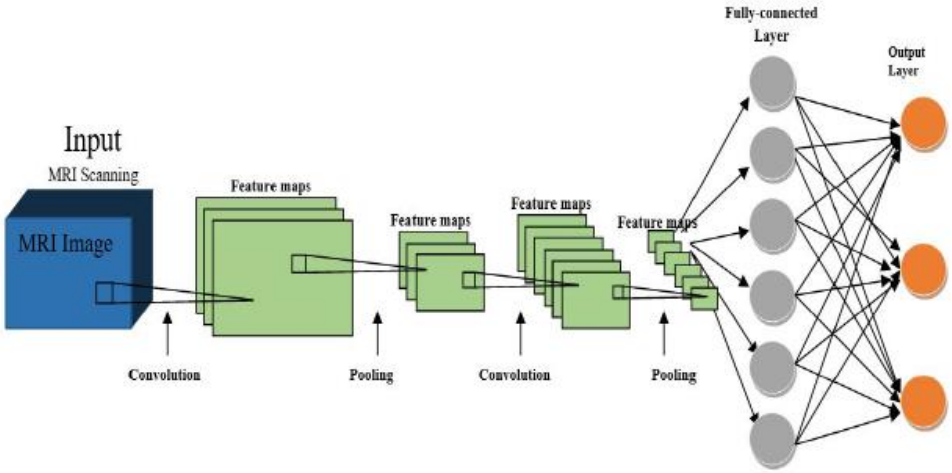
    - Continuously monitor and evaluate the system's performance and user feedback.
    - Incorporate user feedback and expert knowledge to refine the system's algorithms and features.
    - Keep up-to-date with advancements in medical research and technology to enhance the system's capabilities.

7. **Integration with Clinical Workflow:**

    - Integrate the diagnostic system with existing clinical workflows and electronic health record systems to facilitate seamless data exchange and decision support.
    - Ensure interoperability with other healthcare systems and devices to enable comprehensive patient care.

# CHAPTER 4

## METHODOLOGY




1. **Data Preparation:**

    - Gather a dataset of labeled images for training the CNN.
    - Split the dataset into training, validation, and testing sets.

2. **Network Architecture Design:**

    - Design the architecture of the CNN, including the number and arrangement of convolutional layers, pooling layers, and fully connected layers.
    - Select appropriate activation functions, such as ReLU (Rectified Linear Unit), for the layers.
    - Determine the input size based on the dimensions of the input images.

3. **Data Preprocessing:**

    - Resize the input images to a consistent size suitable for the CNN architecture.
    - Normalize the pixel values to a common range, typically between 0 and 1.
    - Apply any additional preprocessing techniques specific to the dataset, such as data augmentation (e.g., rotation, flipping, or scaling) to increase the diversity of the training data.

4. **Convolutional and Pooling Layers:**

    - Pass the preprocessed images through the convolutional layers of the CNN.
    - Apply convolutional filters to extract local features from the input images.
    - Use pooling layers, such as max pooling, to downsample and reduce the spatial dimensions of the feature maps, while retaining the most relevant information.

5. **Fully Connected Layers:**

    - Flatten the output of the last convolutional layer to create a vector representation of the features.
    - Connect the flattened features to one or more fully connected layers.
    - Apply appropriate activation functions to the fully connected layers to introduce non-linearity.

6. **Loss Function and Optimization:**

    - Select an appropriate loss function based on the nature of the classification or regression problem.
    - Common loss functions for classification tasks include cross-entropy loss and softmax activation.
    - Use an optimization algorithm, such as stochastic gradient descent (SGD) or Adam, to update the weights and minimize the loss during training.

7. **Training:**

    - Initialize the weights of the CNN randomly or using pre-trained weights from another model.
    - Feed the training images through the network and calculate the loss.
    - Backpropagate the gradients to update the weights using the selected optimization algorithm.
    - Repeat the process for multiple epochs, adjusting the learning rate and other hyperparameters as needed.
    - Monitor the performance on the validation set to avoid overfitting and make adjustments if necessary.

8. **Evaluation and Testing:**

    - Assess the performance of the trained CNN on the testing set to evaluate its accuracy and generalization capabilities.
    - Calculate evaluation metrics such as accuracy, precision, recall, and F1 score to measure the model's performance.
    - Perform any necessary fine-tuning or model refinement based on the evaluation results.

9. **Prediction:**

    - Use the trained CNN to make predictions on new, unseen images.
    - Apply any necessary post-processing steps, such as thresholding or non-maximum suppression, depending on the specific task and application.

# CHAPTER 5

## BREAST CANCER DETECTION

Breast cancer is a type of cancer that forms in the cells of the breasts. It is the most common cancer among women worldwide, although it can also affect men. Breast cancer typically begins in the milk ducts or the lobules, which are the glands that produce milk.

**Causes:**
The exact causes of breast cancer are still not fully understood. However, certain risk factors have been identified that may increase the likelihood of developing the disease. These risk factors include:

- Age: The risk of breast cancer increases as you get older, with most cases occurring in women over the age of 50.
- Gender: Women are much more likely to develop breast cancer compared to men.
- Family history: Having a close relative, such as a mother, sister, or daughter, who has had breast cancer increases your risk.
- Genetic mutations: Inherited gene mutations, such as BRCA1 and BRCA2, can significantly increase the risk of breast cancer.
- Hormonal factors: Prolonged exposure to estrogen, a hormone that regulates the development and maintenance of female characteristics, can increase the risk. Factors that contribute to increased estrogen exposure include early menstruation, late menopause, and not having children or having them later in life.
- Lifestyle factors: Certain lifestyle choices, such as excessive alcohol consumption, smoking, obesity, and a lack of physical activity, have been associated with an increased risk of breast cancer.

**Symptoms:**
The symptoms of breast cancer can vary, but some common signs to watch for include:

- A lump or thickening in the breast or underarm area.
- Changes in breast size or shape.
- Swelling or redness of the breast.
- Dimpling or puckering of the skin on the breast.
- Nipple changes, such as inversion, scaling, or discharge.
- Persistent pain or tenderness in the breast.

It's important to note that many breast changes are not cancerous, but if you notice any unusual changes in your breasts, it's advisable to consult a healthcare professional for further evaluation.

**Diagnosis and Treatment:**
If breast cancer is suspected, various diagnostic tests may be performed, including mammography, ultrasound, MRI, and biopsy. Once a diagnosis is confirmed, the treatment plan will depend on several factors, such as the stage of cancer, its aggressiveness, and the patient's overall health.

Treatment options for breast cancer may include:

- Surgery: This involves removing the cancerous tumor and possibly nearby lymph nodes. Types of surgery may include lumpectomy (removal of the tumor), mastectomy (removal of the breast tissue), or lymph node removal.
- Radiation therapy: High-energy radiation is used to kill cancer cells or reduce the size of tumors.
- Chemotherapy: Powerful drugs are administered either intravenously or orally to destroy cancer cells throughout the body.
- Hormone therapy: Certain breast cancers are hormone receptor-positive, which means they rely on hormones like estrogen to grow. Hormone therapy involves medications that block the effects of hormones or lower hormone levels in the body.
- Targeted therapy: These medications specifically target certain characteristics of cancer cells to inhibit their growth and spread.

The treatment plan is tailored to each individual and may involve a combination of these approaches. It's essential to discuss the available options with healthcare professionals to make informed decisions about the most suitable treatment for a particular case. Early detection and advances in treatment have significantly improved the survival rates for breast cancer. Regular self-examinations, mammograms, and clinical screenings are important for early detection and better outcomes.

# CHAPTER 6

## BRAIN TUMOR DETECTION

Brain tumours are abnormal cell growths in the brain or the structures nearby. They may be malignant (cancerous) or benign (non-cancerous), and either type can have detrimental effects on a patient's health and wellbeing. For prompt treatment and better patient outcomes, it is essential to identify brain tumours early and correctly.

There are several types of brain tumors, which can be categorized based on their origin, behavior, and cell characteristics. Here are some of the main types:

1. Glioma: Gliomas are the most common type of brain tumors and originate from the glial cells that support and nourish the neurons. They can be further classified into subtypes, including astrocytomas, oligodendrogliomas, and ependymomas.

2. Meningiomas: Meningiomas develop from the meninges, which are the protective membranes covering the brain and spinal cord. They are usually slow-growing and often benign, but they can cause symptoms if they press on surrounding structures.

3. Pituitary Adenomas: Pituitary adenomas are tumors that form in the pituitary gland, which is located at the base of the brain. They can cause hormonal imbalances and various symptoms depending on the hormones affected.

Brain tumour detection is the process of determining whether a brain tumour is present, where it is located, how big it is, and other relevant details. In order to correctly identify and categorise brain tumours, a variety of medical procedures and diagnostic equipment are used.

In recent years, advancements in medical imaging techniques and machine learning algorithms have revolutionized the field of brain tumor detection. Medical professionals and researchers have developed computer-aided diagnosis (CAD) systems that utilize sophisticated algorithms to analyze medical images and assist in the detection and classification of brain tumors.

These CAD systems employ various imaging modalities such as magnetic resonance imaging (MRI), computed tomography (CT), and positron emission tomography (PET) scans to capture detailed images of the brain. These images are then processed and analyzed using machine learning algorithms to identify abnormal regions that may indicate the presence of a brain tumor.

# CHAPTER 7

## CODING AND TESTING

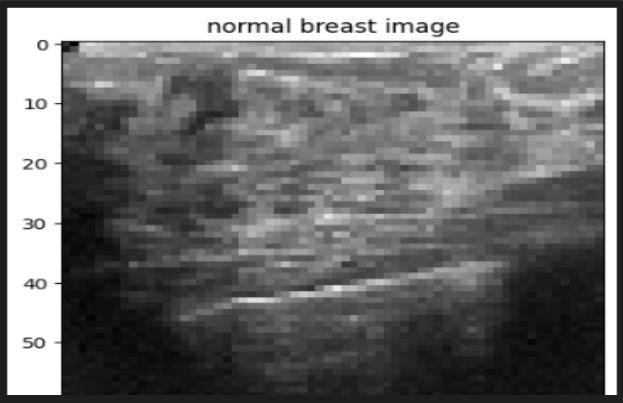
**Breast Cancer Detection:**

### 3.Visualizing the data

#### Normal breast image & mask

```
plt.figure()
plt.imshow(data_set_normal[0])
plt.title("normal breast image")
plt.savefig('normal.jpg',bbox_inches='tight', dpi=150)
plt.figure()
plt.imshow(data_set_normal[1])
plt.title("normal breast mask ")
plt.savefig('normalMask.jpg',bbox_inches='tight', dpi=150)
plt.figure()
histr = cv2.calcHist([data_set_normal[0]],[0],None,[256],[0,256])
plt.plot(histr)
plt.title("Frequency of pixels in an Gray Scale Image")
plt.savefig('NormalPixelFreq.jpg',bbox_inches='tight', dpi=250)
plt.show()
```

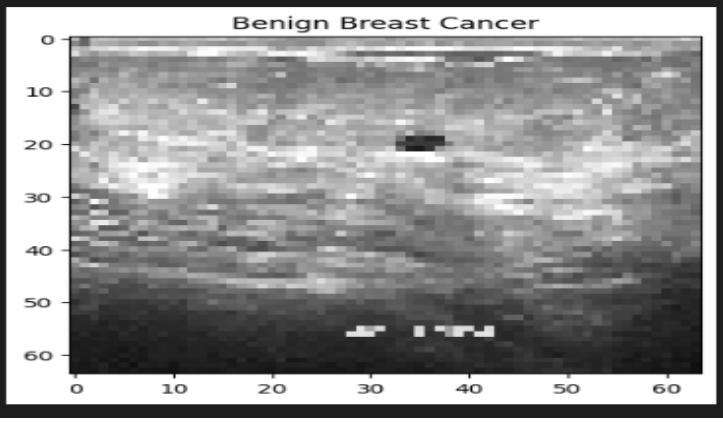
[16]




#### Benign breast cancer & Mask

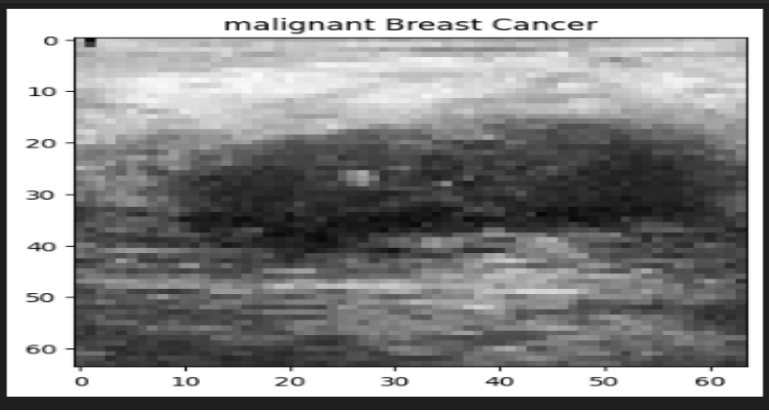
```
plt.figure()
plt.imshow(data_set_benign[0])
plt.title("Benign Breast Cancer")
plt.savefig('beingn.jpg',bbox_inches='tight', dpi=150)
plt.figure()
plt.imshow(data_set_benign[1])
plt.title("Benign Breast Cancer Mask")
plt.savefig('beingnMask.jpg',bbox_inches='tight', dpi=150)
plt.figure()
histr1 = cv2.calcHist([data_set_benign[0]],[0],None,[256],[0,256])
plt.plot(histr1)
plt.title("Frequency of pixels in an Gray Scale Image")
plt.savefig('benignFreqPixels',bbox_inches='tight', dpi=150)
plt.show()
```

[7]



## Malignant breast cancer & Mask

```
plt.figure()
plt.imshow(data_set_malignant[0])
plt.title("malignant Breast Cancer")
plt.savefig('malignant.jpg',bbox_inches='tight', dpi=150)
plt.figure()
plt.imshow(data_set_malignant[1])
plt.title("Malignant Breast Cancer Mask")
plt.savefig('malignantMask.jpg',bbox_inches='tight', dpi=150)
plt.figure()
histr2 = cv2.calcHist([data_set_malignant[0]],[0],None,[256],[0,256])
plt.plot(histr2)
plt.title("Frequency of pixels in an Gray Scale Image")
plt.savefig('MAlignantFreqPixels.jpg',bbox_inches='tight', dpi=150)
plt.show()
```



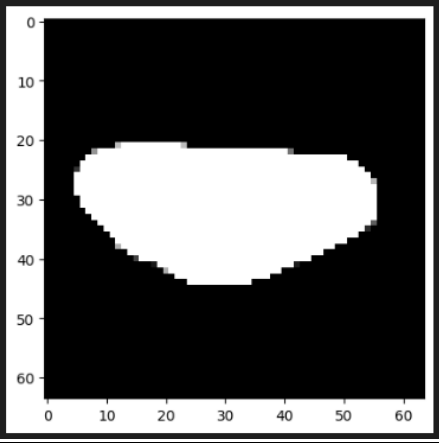

## 3.Splitting the data into training and testing data using scikit-learn

```
pip install scikit-learn
```

```
Collecting scikit-learn
  Downloading scikit_learn-1.2.2-cp39-cp39-win_amd64.whl (8.4 MB)
     ---------------------------------------- 8.4/8.4 MB 5.5 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17.3 in c:\users\sahil\anaconda3\envs\test\lib\site-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.3.2 in c:\users\sahil\anaconda3\envs\test\lib\site-packages (from scikit-learn) (1.10.1)
Collecting threadpoolctl>=2.0.0
  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
Collecting joblib>=1.1.1
  Using cached joblib-1.2.0-py3-none-any.whl (297 kB)
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.2.0 scikit-learn-1.2.2 threadpoolctl-3.1.0
Note: you may need to restart the kernel to use updated packages.
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(data_set_all, targets, test_size=0.3, random_state=42)
```

```
#show an image of the training data to make sure there is nothing wrong
plt.imshow(x_train[0])
plt.show()
y_train[0]
x_train[1000].shape
```



# 4.Building the model

```
model=Sequential()

# 1st conv2D layer
model.add(Conv2D(64,(3,3), padding="same",input_shape=(64,64,3)))
model.add(Activation("relu"))


# 2st conv2D layer
model.add(Conv2D(64,(3,3), padding="same",input_shape=(64,64,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

# 3rd conv2D layer
model.add(Conv2D(128,(3,3)))
model.add(Activation("relu"))



# 4th conv2D layer
model.add(Conv2D(128,(3,3), padding="same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

#5th
model.add(Conv2D(256,(3,3), padding="same"))
model.add(Activation("relu"))
```

```
#6th
model.add(Conv2D(256,(3,3), padding="same"))
model.add(Activation("relu"))

#7th
model.add(Conv2D(256,(3,3), padding="same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))


#8th
model.add(Conv2D(512,(3,3), padding="same"))
model.add(Activation("relu"))

#9th
model.add(Conv2D(512,(3,3), padding="same"))
model.add(Activation("relu"))

#10th
model.add(Conv2D(512,(3,3), padding="same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))


#falttenning
model.add(Flatten())

#dense layer
model.add(Dense(4096))
model.add(Activation('relu'))

#regularization layer
model.add(Dropout(0.5))

#output layer
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

```
[44]
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
history=model.fit(x_train,y_train,batch_size=100,verbose=1,epochs=20
                  , validation_data=(x_test,y_test),
                  shuffle=False)
# plot the history
#tr_plot(history,0)
```

```
Epoch 1/20
12/12 [==============================] - 26s 2s/step - loss: 10.2100 - accuracy: 0.7754 - val_loss: 0.5200 - val_accuracy: 0.8376
Epoch 2/20
12/12 [==============================] - 25s 2s/step - loss: 0.3219 - accuracy: 0.8804 - val_loss: 0.2596 - val_accuracy: 0.9262
Epoch 3/20
12/12 [==============================] - 26s 2s/step - loss: 0.2657 - accuracy: 0.9112 - val_loss: 0.2586 - val_accuracy: 0.9262
Epoch 4/20
12/12 [==============================] - 28s 2s/step - loss: 0.2269 - accuracy: 0.9112 - val_loss: 0.1994 - val_accuracy: 0.9262
Epoch 5/20
12/12 [==============================] - 28s 2s/step - loss: 0.1932 - accuracy: 0.9112 - val_loss: 0.2210 - val_accuracy: 0.9262
Epoch 6/20
12/12 [==============================] - 28s 2s/step - loss: 0.1907 - accuracy: 0.9139 - val_loss: 0.1789 - val_accuracy: 0.9262
Epoch 7/20
12/12 [==============================] - 28s 2s/step - loss: 0.1805 - accuracy: 0.9112 - val_loss: 0.1690 - val_accuracy: 0.9262
Epoch 8/20
12/12 [==============================] - 28s 2s/step - loss: 0.1659 - accuracy: 0.9176 - val_loss: 0.1728 - val_accuracy: 0.9283
Epoch 9/20
12/12 [==============================] - 30s 2s/step - loss: 0.1456 - accuracy: 0.9330 - val_loss: 0.1871 - val_accuracy: 0.9114
Epoch 10/20
12/12 [==============================] - 28s 2s/step - loss: 0.1515 - accuracy: 0.9203 - val_loss: 0.1656 - val_accuracy: 0.9262
Epoch 11/20
12/12 [==============================] - 28s 2s/step - loss: 0.1369 - accuracy: 0.9257 - val_loss: 0.1741 - val_accuracy: 0.9262
Epoch 12/20
12/12 [==============================] - 28s 2s/step - loss: 0.1178 - accuracy: 0.9239 - val_loss: 0.2150 - val_accuracy: 0.9367
Epoch 13/20
12/12 [==============================] - 28s 2s/step - loss: 0.1829 - accuracy: 0.9230 - val_loss: 0.2525 - val_accuracy: 0.9262
Epoch 14/20
12/12 [==============================] - 28s 2s/step - loss: 0.1649 - accuracy: 0.9221 - val_loss: 0.1388 - val_accuracy: 0.9325
Epoch 15/20
12/12 [==============================] - 27s 2s/step - loss: 0.1203 - accuracy: 0.9366 - val_loss: 0.2055 - val_accuracy: 0.9325
Epoch 16/20
12/12 [==============================] - 28s 2s/step - loss: 0.1480 - accuracy: 0.9149 - val_loss: 0.1927 - val_accuracy: 0.9430
Epoch 17/20
12/12 [==============================] - 28s 2s/step - loss: 0.1598 - accuracy: 0.9266 - val_loss: 0.1810 - val_accuracy: 0.9388
Epoch 18/20
12/12 [==============================] - 28s 2s/step - loss: 0.1368 - accuracy: 0.9375 - val_loss: 0.1388 - val_accuracy: 0.9325
Epoch 19/20
12/12 [==============================] - 28s 2s/step - loss: 0.2793 - accuracy: 0.9357 - val_loss: 0.1683 - val_accuracy: 0.9241
Epoch 20/20
12/12 [==============================] - 29s 2s/step - loss: 0.1725 - accuracy: 0.9085 - val_loss: 0.1922 - val_accuracy: 0.9030
```

**Brain Tumour Detection:**

## 2. Storing the data

```
import os
X_train = []
Y_train = []
image_size = 150
labels = ['glioma_tumor','meningioma_tumor','no_tumor','pituitary_tumor']
for i in labels:
    folderPath = os.path.join('Training',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)

for i in labels:
    folderPath = os.path.join('Testing',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)

X_train = np.array(X_train)
Y_train = np.array(Y_train)
```

```
X_train,Y_train = shuffle(X_train,Y_train,random_state=101)
X_train.shape
```

```
(3264, 150, 150, 3)
```

## 3. Splitting of data into training and testing

```
X_train,X_test,y_train,y_test = train_test_split(X_train,Y_train,test_size=0.2,random_state=101)
```

## 4. Pre-processing of Images

```
y_train_new = []
for i in y_train:
    y_train_new.append(labels.index(i))
y_train=y_train_new
y_train = tf.keras.utils.to_categorical(y_train)

y_test_new = []
for i in y_test:
    y_test_new.append(labels.index(i))
y_test=y_test_new
y_test = tf.keras.utils.to_categorical(y_test)
```

# 5. Building the Model

```
model = Sequential()
model.add(Conv2D(32,(3,3),activation = 'relu',input_shape=(150,150,3)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Dropout(0.3))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(256,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(512,activation = 'relu'))
model.add(Dense(512,activation = 'relu'))
model.add(Dropout(0.3))
model.add(Dense(4,activation='softmax'))
```

# 6. Compiling and Training the Model

```
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

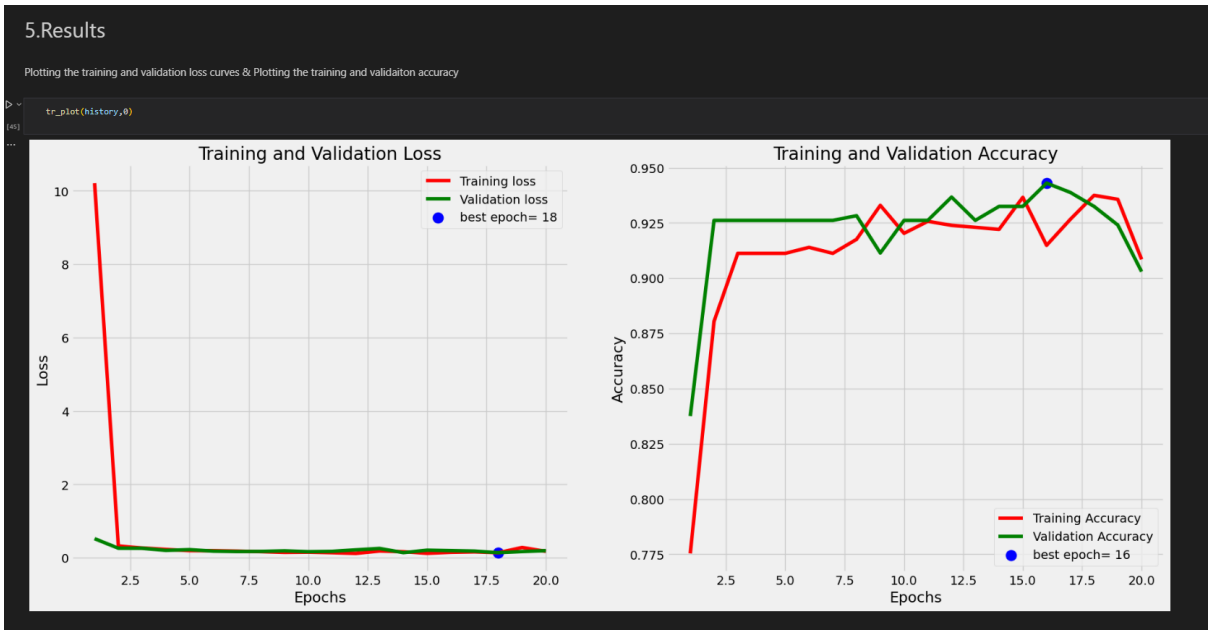
```
history = model.fit(X_train,y_train,epochs=20,validation_split=0.1)
```

```
Epoch 1/20
74/74 [==============================] - 153s 2s/step - loss: 1.9528 - accuracy: 0.2750 - val_loss: 1.3688 - val_accuracy: 0.2901
Epoch 2/20
74/74 [==============================] - 169s 2s/step - loss: 1.3508 - accuracy: 0.2920 - val_loss: 1.3649 - val_accuracy: 0.2901
Epoch 3/20
74/74 [==============================] - 166s 2s/step - loss: 1.2892 - accuracy: 0.3951 - val_loss: 1.2177 - val_accuracy: 0.4618
Epoch 4/20
74/74 [==============================] - 172s 2s/step - loss: 1.1142 - accuracy: 0.5313 - val_loss: 1.0533 - val_accuracy: 0.5191
Epoch 5/20
74/74 [==============================] - 156s 2s/step - loss: 0.9032 - accuracy: 0.5994 - val_loss: 1.1102 - val_accuracy: 0.5305
Epoch 6/20
74/74 [==============================] - 147s 2s/step - loss: 0.8064 - accuracy: 0.6411 - val_loss: 0.8089 - val_accuracy: 0.6221
Epoch 7/20
74/74 [==============================] - 149s 2s/step - loss: 0.7276 - accuracy: 0.6897 - val_loss: 0.8203 - val_accuracy: 0.6183
Epoch 8/20
74/74 [==============================] - 271s 4s/step - loss: 0.6594 - accuracy: 0.7275 - val_loss: 0.7125 - val_accuracy: 0.6794
Epoch 9/20
74/74 [==============================] - 372s 5s/step - loss: 0.5879 - accuracy: 0.7620 - val_loss: 0.6506 - val_accuracy: 0.7290
Epoch 10/20
74/74 [==============================] - 420s 6s/step - loss: 0.5357 - accuracy: 0.7961 - val_loss: 0.6554 - val_accuracy: 0.6908
Epoch 11/20
74/74 [==============================] - 281s 4s/step - loss: 0.4495 - accuracy: 0.8250 - val_loss: 0.6426 - val_accuracy: 0.7137
Epoch 12/20
74/74 [==============================] - 229s 3s/step - loss: 0.3984 - accuracy: 0.8497 - val_loss: 0.5431 - val_accuracy: 0.7748
Epoch 13/20
74/74 [==============================] - 423s 6s/step - loss: 0.3290 - accuracy: 0.8753 - val_loss: 0.4874 - val_accuracy: 0.8244
Epoch 14/20
74/74 [==============================] - 386s 5s/step - loss: 0.2985 - accuracy: 0.8876 - val_loss: 0.4916 - val_accuracy: 0.8168
Epoch 15/20
74/74 [==============================] - 280s 4s/step - loss: 0.2625 - accuracy: 0.9042 - val_loss: 0.4469 - val_accuracy: 0.8244
Epoch 16/20
74/74 [==============================] - 264s 4s/step - loss: 0.2421 - accuracy: 0.9097 - val_loss: 0.4172 - val_accuracy: 0.8588
Epoch 17/20
74/74 [==============================] - 289s 4s/step - loss: 0.2274 - accuracy: 0.9149 - val_loss: 0.3969 - val_accuracy: 0.8550
Epoch 18/20
74/74 [==============================] - 279s 4s/step - loss: 0.2378 - accuracy: 0.9127 - val_loss: 0.4239 - val_accuracy: 0.8397
Epoch 19/20
74/74 [==============================] - 280s 4s/step - loss: 0.1625 - accuracy: 0.9383 - val_loss: 0.3944 - val_accuracy: 0.8702
Epoch 20/20
74/74 [==============================] - 276s 4s/step - loss: 0.1448 - accuracy: 0.9476 - val_loss: 0.4433 - val_accuracy: 0.8626
```

# CHAPTER 8

# EVALUATION AND RESULTS

## Breast Cancer Detection:



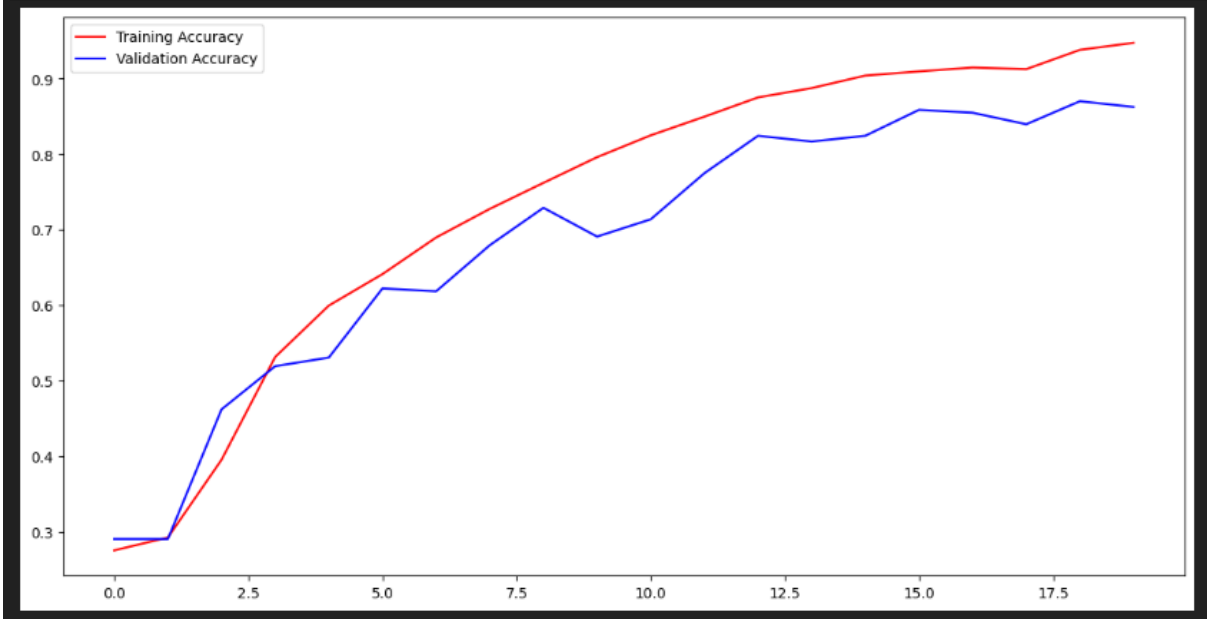

0.9714285714285714

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 2 | 0.98 | 0.98 | 0.98 | 85 |
| 4 | 0.96 | 0.96 | 0.96 | 55 |
| accuracy | | | 0.97 | 140 |
| macro avg | 0.97 | 0.97 | 0.97 | 140 |
| weighted avg | 0.97 | 0.97 | 0.97 | 140 |

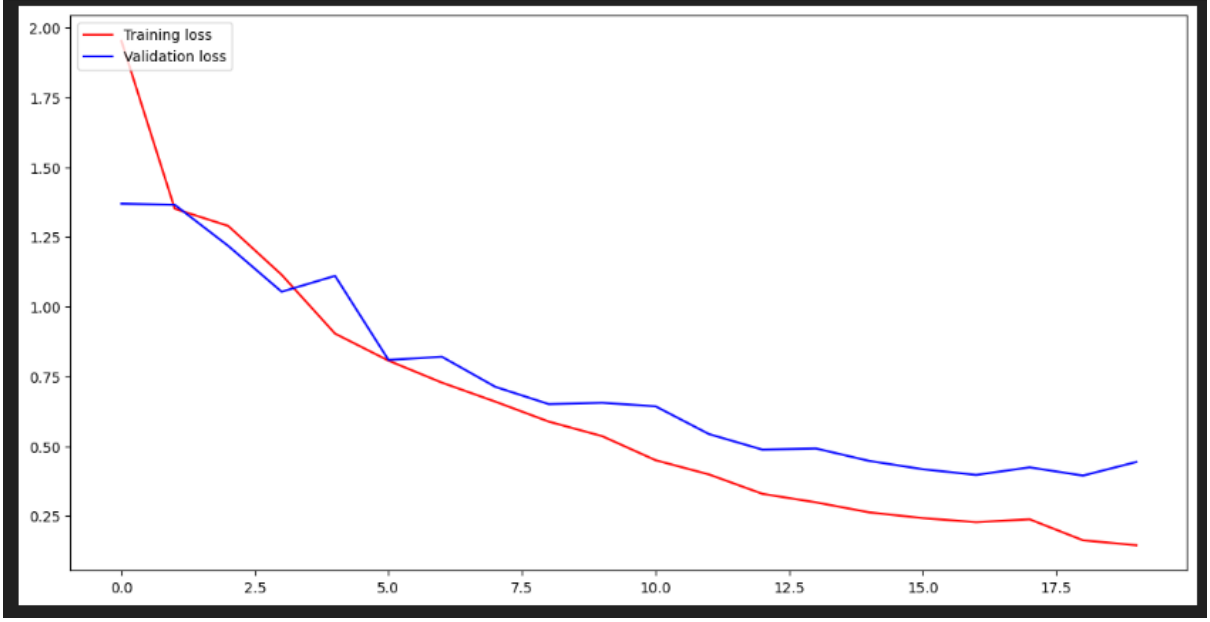
## Brain Tumour Detection:

# 7. Model Evaluation

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epochs = range(len(acc))
fig = plt.figure(figsize=(14,7))
plt.plot(epochs,acc,'r',label="Training Accuracy")
plt.plot(epochs,val_acc,'b',label="Validation Accuracy")
plt.legend(loc='upper left')
plt.show()
```




```
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
fig = plt.figure(figsize=(14,7))
plt.plot(epochs,loss,'r',label="Training loss")
plt.plot(epochs,val_loss,'b',label="Validation loss")
plt.legend(loc='upper left')
plt.show()
```



# Prediction

## 0 - GLIOMA TUMOR, 1 - MENINGIOMA TUMOR , 2- NO TUMOR, 3- PITUITARY TUMOR

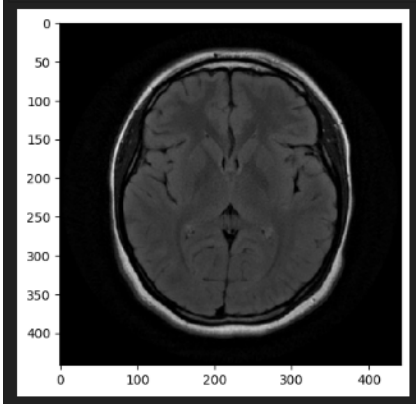
```
img = cv2.imread("C:\\Users\\win10\\OneDrive\\Desktop\\Projects\\Brain_Tumor_Detection_new\\Training\\no_tumor\\image (27).jpg")
img = cv2.resize(img,(150,150))
img_array = np.array(img)
img_array.shape
```

```
(150, 150, 3)
```

```
img_array = img_array.reshape(1,150,150,3)
img_array.shape
```

```
(1, 150, 150, 3)
```

```
from tensorflow.keras.preprocessing import image
img = image.load_img("C:\\Users\\win10\\OneDrive\\Desktop\\Projects\\Brain_Tumor_Detection_new\\Training\\no_tumor\\image (27).jpg")
plt.imshow(img,interpolation='nearest')
plt.show()
```




```
a=model.predict(img_array)
indices = a.argmax()
indices
```

```
1/1 [==============================] - 0s 31ms/step

2
```

# CHAPTER 9

## CONCLUSION AND FUTURE ENHANCEMENT

The medical diagnostics project has shown how machine learning algorithms and modern technologies can be used to increase diagnostic efficiency and accuracy. A system that makes use of a variety of datasets and machine learning models to aid in the identification of various medical disorders was successfully established as part of the project.

The study demonstrated the promise of data-driven techniques in healthcare through significant investigation and experimentation. The system's success in correctly identifying and forecasting medical issues emphasises how valuable it is as an additional tool for healthcare providers.

The deployment of the medical diagnosis system may benefit healthcare in a number of ways. It might lessen diagnostic blunders, improve patient outcomes, and maximise resource use. The technology can enhance the decision-making processes of healthcare providers and help to create more individualized and efficient treatment plans by giving them useful insights.

The medical diagnosing system still has room for improvement and evolution in the future. The accuracy and reliability of the system must be continuously improved, and this includes honing machine learning algorithms and incorporating fresh medical data. To keep the system current with the most recent medical knowledge and procedures, ongoing contact with medical specialists is essential.

The system's capabilities can also be improved by combining it with electronic health records (EHRs) and other healthcare information systems. Through seamless access to patient data, real-time updates, and a thorough understanding of the patient's medical history, more informed and precise diagnoses would be possible.
Future developments of the medical diagnosing system must also take ethical issues and privacy concerns into account. Stricter data security measures and adherence to privacy regulations are necessary to protect patient confidentiality and ensure trust in the system.

In conclusion, the medical diagnosis project has showcased the potential of data-driven approaches and machine learning in improving medical diagnosis. With further enhancements and advancements, the system can contribute to more accurate and efficient healthcare delivery

# CHAPTER 10

## REFERENCES


1. https://www.researchgate.net/publication/347334482_Medical_Diagnostic_Systems_Using_Artificial_Intelligence_AI_Algorithms_Principles_and_Perspectives
2. McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., ... & Darzi, A. (2020). International evaluation of an AI system for breast cancer screening. Nature, 577(7788), 89-94.
3. Obermeyer, Z., & Emanuel, E. J. (2016). Predicting the future—big data, machine learning, and clinical medicine. New England Journal of Medicine, 375(13), 1216-1219.
4. https://iopscience.iop.org/article/10.1088/1757-899X/1055/1/012115
5. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-023-02114-6
6. https://www.sciencedirect.com/science/article/pii/S2667102621000887
7. https://arxiv.org/ftp/arxiv/papers/2203/2203.04308.pdf
8. https://medium.datadriveninvestor.com/breast-cancer-detection-using-machine-learning-475d3b63e18e